Available online at www.sciencedirect.com
ScienceDirect
Discrete Applied Mathematics 155 (2007) 1408–1419
DISCRETE APPLIED MATHEMATICS
www.elsevier.com/locate/dam

Satisfiability of mixed Horn formulas☆

Stefan Porschen, Ewald Speckenmeyer

Institut für Informatik, Universität zu Köln, D-50969 Köln, Germany

Received 8 February 2005; received in revised form 9 February 2007; accepted 18 February 2007
Available online 4 March 2007

Abstract

In this paper the class of *mixed Horn formulas* is introduced that contain a Horn part and a 2-CNF (conjunctive normal form) (also called quadratic) part. We show that SAT remains NP-complete for such instances and also that any CNF formula can be encoded in terms of a mixed Horn formula in polynomial time. Further, we provide an exact deterministic algorithm showing that SAT for mixed Horn formulas containing n variables is solvable in time $O(2^{0.5284n})$. A strong argument showing that it is hard to improve a time bound of $O(2^{n/2})$ for mixed Horn formulas is provided. We also obtain a fixed-parameter tractability classification for SAT restricted to mixed Horn formulas C of at most k variables in its positive 2-CNF part providing the bound $O(\|C\|2^{0.5284k})$. We further show that the NP-hard optimization problem minimum weight SAT for mixed Horn formulas can be solved in time $O(2^{0.5284n})$ if non-negative weights are assigned to the variables. Motivating examples for mixed Horn formulas are level graph formulas [B. Randerath, E. Speckenmeyer, E. Boros, P. Hammer, A. Kogan, K. Makino, B. Simeone, O. Cepek, A satisfiability formulation of problems on level graphs, ENDM 9 (2001)] and graph colorability formulas.

Keywords: (Hidden) Horn formula; q-Horn formula; Quadratic formula; (Weighted) Satisfiability; Minimal vertex cover; Fixed-parameter tractability

1. Introduction

In recent time the interest in designing exact algorithms providing better upper time bounds than the trivial ones for NP-complete problems and their NP-hard optimization counterparts has increased. Of particular interest in this context is the investigation of exact algorithms for testing the satisfiability (SAT) of propositional formulas in conjunctive normal form (CNF). This interest stems from the fact that SAT is well known to be a fundamental NP-complete problem appearing naturally or via reduction as the abstract core of many application-relevant problems. Not only the whole class CNF is of interest in this context. In several applications subclasses of CNF are of importance for which SAT unfortunately remains NP-complete. Nevertheless, it is often possible by exploiting the specific structure of such formulas to design fast exact algorithms for their solution. Such subclasses, for instance, can be obtained by composing or *mixing* formulas of two different parts each of which separately is SAT-testable in polynomial time (see also [12]).

In this paper we introduce and study so-called *mixed Horn formulas* which roughly speaking are formulas composed of a quadratic part and a Horn part. More precisely, for a positive monotone 2-CNF formula P (containing only 2-clauses) and a Horn formula H, we call the formula $M = H \wedge P$ a *mixed Horn formula* (*MHF*). It is well known

☆ A preliminary version of this paper appeared in [16].

E-mail addresses: porschen@informatik.uni-koeln.de (S. Porschen), esp@informatik.uni-koeln.de (E. Speckenmeyer).

0166-218X/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2007.02.010

that 2-SAT and Horn-SAT are solvable in linear time [1,14], but SAT for MHFs (MHF-SAT) remains NP-complete. A closely related class generalizing the classes of Horn and quadratic formulas is the class of so-called q-Horn formulas introduced by Boros et al. [2], for which SAT can be solved in linear time also [2]. A q-Horn formula (in partition form, see below), can be considered as a specific mixed Horn formula. The class of q-Horn formulas (in partition form), probably is the largest subclass of MHF that is SAT-solvable in polynomial time.

The main purpose of this paper is to prove a non-trivial worst case upper time bound for solving MHF-SAT, namely $O(2^{0.5284n})$ where n is the number of variables in the instance. Moreover, we obtain a fixed-parameter tractability classification (cf. e.g. [5]) of SAT restricted to MHFs $M = P \wedge H$ where P has a fixed number k of different variables, provided by the polynomial bound $O(\|M\| 2^{0.5284k})$, where $\|M\|$ is the length of M.

We also analyze the connection of MHF-SAT to unrestricted SAT. Specifically we show that each CNF formula C with n different variables can be transformed in polynomial time into a MHF $M = P \wedge H$, such that P has $k \leqslant 2n$ different variables. Then C is satisfiable if and only if M is satisfiable, and the question, whether $M \in$ SAT, can be answered in time $O(\|C\| 2^{k/2})$. Hence, if there is an $\alpha < \frac{1}{2}$ such that every MHF $M = P \wedge H$ can be solved in time $O(\|C\| 2^{\alpha k})$, then there is $\beta \leqslant 2\alpha < 1$ such that SAT for an arbitrary CNF-formula C can be decided in time $O(\|C\| 2^{\beta n})$. The MHF-formulation of a CNF-formula C yields a partition of all variables in C into the *essential* variables (variables occurring in P) and the remaining ones.

The introduction and investigation of MHFs is by no means artificial. Well known problems for level graphs, like level-planarity test or the NP-hard crossing-minimization problem, can be formulated conveniently in terms of MHFs (for more details see [17]). This was our motivation for considering MHFs. Also graph colorability naturally leads to MHFs. To see this, consider a simple graph $G = (V, E)$ and a set of r colors $[r] := \{1, \ldots, r\}$. The decision whether G is r-colorable, i.e. whether at most r colors can be assigned to all vertices in V such that no two adjacent vertices are colored equally, can be encoded into MHF-SAT as follows: for every vertex $x \in V$ introduce r variables x_i, $i \in [r]$, and one clause $x_1 \vee x_2 \vee \cdots \vee x_r$. For every edge $x - y \in E$, we have to ensure that x and y are colored differently. So we introduce for each color $i \in [r]$ the clause $\neg(x_i \wedge y_i) \equiv (\overline{x_i} \vee \overline{y_i})$ yielding r 2-clauses for each edge. In summary, we obtain a CNF formula $C(G)$ consisting of $|V| + r|E|$ clauses and containing $r|V|$ different variables. Finally complementing all variables in $C(G)$ turns all its r-clauses into negative monotone clauses and its 2-clauses into positive monotone clauses, hence yields a MHF $\tilde{C}(G)$. It is easy to verify that G is r-colorable if and only if the MHF $\tilde{C}(G)$ is satisfiable via the interpretation that setting variable x_i to FALSE means that the corresponding vertex x is colored by i. Notice that introducing only one r-clause for every vertex ensuring at least (instead of exactly) one color for every vertex suffices for deciding r-colorability.

Another source of the interest in Horn clauses contained in CNF formulas stems from recent observations of hidden threshold phenomena [20] according to a fixed fraction of Horn clauses in CNF formulas.

The present paper is structured as follows. Section 2 introduces basic notions and notations used throughout the paper. In Section 3, several versions of MHFs are introduced. Each of these classes is NP-complete w.r.t. SAT as follows by the above described reduction from the NP-complete graph coloring problem [6]. We provide another polynomial time transformation of CNF-SAT to MHF-SAT on which some investigations in this paper rely. In Section 4, a vertex cover based algorithm for determining SAT of a MHF M is presented having running time $O(2^{0.5284n})$, with n being the number of variables in M. The approach also yields a classification of MHFs allowing for a fixed-parameter tractability result. Section 5 provides a strong argument stating that it is hard to improve an $O(2^{n/2})$ time bound for solving MHF-SAT. Section 6, describes a further vertex cover based technique for speeding up the MHF-SAT algorithm. Some experimental results illustrating the usefulness of this approach are presented. Section 7, finally, provides an algorithm for the minimum weight MHF-SAT problem, where weights are assigned to the variables.

2. Basic notions and notation

Let CNF denote the set of formulas (free of duplicate clauses) in CNF over a set $V = \{x_1, \ldots, x_n\}$ of propositional variables $x_i \in \{0, 1\}$. Each variable x induces a positive literal (variable x) or a negative literal (negated variable: $\overline{x}$). Each formula $C \in$ CNF is considered as a clause set $C = \{c_1, \ldots, c_{|C|}\}$. Each clause $c \in C$ is a disjunction of different literals, and is also represented as a set $c = \{l_1, \ldots, l_{|c|}\}$. The length of a formula C is denoted by $\|C\|$ whereas $|C|$ denotes the number of its clauses. A clause containing positive (negative) literals only is called *positive* (*negative*) *monotone*. We denote by $V(C)$ the set of variables occurring in formula C. The satisfiability problem (SAT) asks in its *decision* version, whether a given CNF instance C is *satisfiable*, i.e. whether C has a *model*, which is a truth assignment

$\tau : V(C) \to \{0, 1\}$ setting at least one literal in each clause of C to 1 (TRUE). For convenience we allow the empty set to be a formula: $\emptyset \in$ CNF which always is satisfiable. In its *search* version SAT, in addition, means to find a model τ if the input formula is satisfiable.

For $C \in$ CNF and a subset $X \subseteq V(C)$ of its variables, we define C^X as the formula obtained from C by flipping each variable in X. Flipping all variables in C is abbreviated by $C^\gamma = C^{V(C)}$. Given a formula $C \in$ CNF and a partial truth assignment $\tau : V(C) \to \{0, 1\}$, we denote by $C[\tau]$ the *reduced formula* obtained from C by removing all clauses satisfied by τ and removing all literals from the remaining clauses which are set to 0 (FALSE) by τ. Obviously, if τ is a model of C then $C[\tau] = \emptyset$. For two partial truth assignments τ, τ_1 of a formula C, satisfying $\tau_1 \subseteq \tau$, i.e. $D(\tau_1) \subseteq D(\tau)$ (for their domains) and $\tau_1^{-1}(1) \subseteq \tau^{-1}(1)$, obviously holds: if τ satisfies C, then $C[\tau_1]$ is satisfiable.

For $k \in \mathbb{N}$, let k-CNF (resp. CNF($=k$)) denote the subset of formulas C such that each clause has a length of at most (resp. exactly) k. Moreover, $\mathscr{M}_\varepsilon, \varepsilon \in \{+, -\}$, denotes the set of ε-*monotone* (CNF-)formulas, i.e. for $\varepsilon = +$ ($-$) all clauses are positive (negative) monotone. Let $\mathscr{H}$ denote the set of all *Horn* formulas, each clause of which has at most one positive literal. For a *hidden* (or *disguised*) Horn formula H, by definition there exists a subset $X \subset V(H)$ such that H^X is a Horn formula. The set of all hidden Horn formulas is denoted by $\hat{\mathscr{H}}$. Moreover, we consider *q-Horn* formulas [2,3] the class of which is denoted as q-$\mathscr{H}$. $C \in$ CNF is q-Horn if there is $X \subseteq V(C)$ such that C^X is in *partition-form*. A formula C is said to be in partition-form if there are $Q, H \subseteq V(C)$ with $V(C) = Q \cup H$, $Q \cap H = \emptyset$, and for every clause $c \in C$, exactly one of the following cases holds: (1) c contains only variables in H and exactly one literal is positive, (2) c contains at most two variables in Q and arbitrary many variables in H appearing negative. Note that every Horn formula is in partition-form with $Q = \emptyset$; also note that $\hat{H} \subset q$-$\mathscr{H}$.

For a monotone formula $C \in \mathscr{M}_\varepsilon$ ($\varepsilon \in \{+, -\}$), we can construct its *formula graph* G_C with vertex set $V(C)$ in linear time. Two vertices are joined by an edge if there is a clause in C containing the corresponding variables. Clearly, for each $c \in C$ the subgraph $G_C|c$ of G_C is isomorphic to the complete graph $K_{|c|}$. In the particular case of $C \in \mathscr{M}_\varepsilon(=2)$, i.e. C is a monotone formula containing 2-clauses only, G_C contains exactly one edge for every clause in C. Note that a monotone formula $C \in$ CNF($=2$) with each variable occurring only once corresponds to a graph of isolated edges only, and whose number of edges is half the number of vertices.

3. Mixing Horn and quadratic formulas

Let $\mathscr{C}_1, \mathscr{C}_2 \subset$ CNF be two classes of formulas over the same variable set V. A formula $C \in$ CNF such that there are formulas $C_i \in \mathscr{C}_i, i = 1, 2$, with $C = C_1 \wedge C_2$, is called *mixed* (*over* $\mathscr{C}_1, \mathscr{C}_2$). The collection of formulas mixed over $\mathscr{C}_1, \mathscr{C}_2$ is denoted as $\mathscr{C}_1 \wedge \mathscr{C}_2$. In this paper we are interested in specific mixed formulas containing Horn subformulas.

Definition 1. We define the class $\mathscr{C}_1 \wedge$ 2-CNF as *negative mixed Horn formulas*, MHF$_-$, if $\mathscr{C}_1 = \mathscr{M}_-$; as *mixed Horn formulas*, MHF, if $\mathscr{C}_1 = \mathscr{H}$; as *mixed hidden Horn formulas* MHHF, if $\mathscr{C}_1 = \hat{\mathscr{H}}$; and as *mixed q-Horn formulas*, MqHF, if $\mathscr{C}_1 = q$-$\mathscr{H}$.

Because all 2-clauses which are not positive monotone are Horn, every formula $M \in$ MHF has the unique representation $M = H \wedge P$, where P is the collection of all positive monotone 2-clauses in M and H is the remaining Horn subformula. Given $M \in$ MHF, we denote these subformulas as $P(M)$ resp. $H(M)$.

The question arises whether the mixed formulas introduced in Definition 1 can be recognized fast. It is obvious that membership of MHF$_-$ and MHF can be recognized in time $O(\|C\|)$, for $C \in$ CNF. The next lemma gives a positive answer also for recognizing mixed hidden Horn and mixed q-Horn formulas. We denote the set of all formulas in partition-form as $\mathscr{2H} \subset q$-$\mathscr{H}$.

Lemma 1. *For* $C \in$ CNF, *it can be decided in time* $O(\|C\|)$ *whether* $C \in$ MHHF (*resp.* $C \in$ MqHF). *Moreover, in linear time*, $C \in$ MHHF (*resp.* $C \in$ MqHF) *can be turned into a* SAT-*equivalent formula* $C' \in$ MHF (*resp.* $C' \in \mathscr{2H} \wedge$ 2-CNF).

Proof. For $C \in$ CNF, let $T := T(C)$ be its largest 2-CNF subformula and let $\hat{C} := C \backslash T$, then obviously $C \in$ MHHF (resp. $C \in$ MqHF) holds iff $\hat{C}$ is hidden Horn (resp. is q-Horn), i.e. iff there is $X \subseteq V(\hat{C})$ such that $\hat{C}^X$ is Horn (resp. $\hat{C}^X$ is in partition-form) from which follows that $C^X = \hat{C}^X \wedge T^X \in$ MHF (resp. $C^X \in \mathscr{2H} \wedge$2-CNF). The test whether such an X exists, and thus, the transformation of C into a SAT-equivalent formula $C^X \in$ MHF (resp. $C^X \in \mathscr{Q}\mathscr{H} \wedge$ 2-CNF), is possible in linear time $O(\|C\|)$ due to [13,3]. $\square$

It is not hard to see that the reduction from graph colorability to MHF-SAT presented in the introduction is in fact a reduction to MHF_--SAT. Using the (proper) inclusions $\text{MHF}_- \subset \text{MHF} \subset \text{MHHF} \subset \text{MqHF}$ we immediately obtain:

Proposition 1. SAT *remains* NP-*complete for each of the classes* MHF_-, MHF, MHHF, *and* MqHF.

Next we describe a transformation of CNF-SAT to MHF-SAT, which is reconsidered in Section 5. This transformation also provides a different look at CNF-SAT solving from the point of view of MHFs.

Transformation (*CNF-To-MHF*).
Input: $\emptyset \neq C \in$ CNF
Output: $M_C \in \text{MHF}_-$, s.t. $M_C \in$ SAT iff $C \in$ SAT.
Let $V_+(C) \subseteq V(C)$ be the set of all variables that occur positive in at least one k-clause of C with $k \geqslant 3$. For every variable $x \in V_+(C)$, introduce a new variable y_x. Then:

(1) Replace all positive occurrences of $x \in V_+(C)$ in the k-clauses $k \geqslant 3$ by $\overline{y}_x$, for every $x \in V_+(C)$. Let the formula obtained be C'.
(2) Add the constraints $\overline{y}_x \leftrightarrow x$ to C', for all $x \in V_+(C)$. This yields the new CNF formula

$$M_C := C' \cup \bigcup_{x \in V_+(C)} \{y_x, x\} \cup \{\overline{y}_x, \overline{x}\}.$$

In the last step we have used the simple equivalences $\overline{y}_x \leftrightarrow x \equiv \overline{y}_x \rightarrow x \wedge x \rightarrow \overline{y}_x$ and $a \rightarrow b \equiv \overline{a} \vee b$. Because all positive literals occurring in k-clauses of C with $k \geqslant 3$ are removed, $M_C \in \text{MHF}_-$ holds.

Transformation CNF-To-MHF obviously consumes polynomial time only and is a reduction in the sense that $C \in$ SAT if and only if $M_C \in$ SAT. It can be adapted also to obtain a MHF that is not necessarily a member of the class MHF_-. For this, it is often not necessary to create for every $x \in V_+(C)$ a new variable as indicated above. A subset of $V_+(C)$, as small as possible, suffices to yield a (not necessarily negative monotone) Horn part and thus may produce a smaller positive monotone part P of 2-clauses. It turns out that the size of P is the crucial quantity regarding the running time of Algorithm MHFSAT described in the next section.

4. Solving SAT for mixed Horn formulas

We aim at providing a non-trivial exact deterministic algorithm solving the SAT search problem for the classes MHF_-, MHF, MHHF, and MqHF. As it turns out, it is convenient to address the class MHF, first. For $M \in$ MHF, we assume that $P := P(M) \in \mathscr{M}_+(=2)$ is not the empty formula. Since otherwise a model for $M = H(M) \in \mathscr{H}$ can be found by Horn-SAT, if existing. Since P is monotone and each of its clauses is a 2-clause, the formula graph G_P of P has exactly one edge for each clause in P, i.e. $G_P = (V(P), P)$. By monotonicity P obviously is satisfiable. Observe that for satisfying P it suffices to find a set of variables X hitting all clauses of P and to set every variable in X to 1. The remaining variables in P are free, i.e. independent of P and if possible should be assigned appropriately to satisfy the remaining Horn formula, too. In terms of the formula graph G_P, such a set X corresponds to a vertex cover of G_P. In other words running through all vertex covers of G_P means running through all models of P. For every such (partial) model of P, we can test by Horn-SAT whether it can be extended to a model of the remaining Horn formula $H(M)$ and thus to a model of the whole instance M. Due to the following observation it is not necessary to test every vertex cover of G_P:

Lemma 2. $M = P \wedge H \in$ MHF *is satisfiable if and only if there exists a minimal vertex cover of* G_P *which can be extended to a model of* M.

Proof. Suppose that $M = H \wedge P \in$ MHF is satisfiable and let σ be a model of M. Then $H \in$ SAT and $\sigma_P := \sigma|V(P)$ is a model of P. Restricting the domain of σ_P to those variables $x \in V(P)$ with $\sigma_P(x) = 1$ also yields a model τ of P with $D(\tau) = \tau^{-1}(1)$, because P is positive monotone. Clearly, the set $X := \{x \in V(P) : \sigma_P(x) = 1\}$ represents a vertex cover of G_P. If X is a minimal vertex cover of G_P we are done. Otherwise, this vertex cover contains a minimal vertex cover of G_P corresponding to a truth assignment τ' that is also a model of P. By construction $D(\tau') = \tau'^{-1}(1) \subset \sigma^{-1}(1)$ holds. Hence, τ' is contained in σ yielding $M[\tau'] \in$ SAT which means that τ' can be extended to a model of M proving the only-if part of the lemma. The converse direction is obvious. □

Therefore, an algorithm that enumerates all minimal vertex covers of G_P and that for each cover separately checks in linear time whether the remaining Horn formula is satisfiable, definitely performs the task of solving SAT for M. It is well known that the complement of a vertex set of a minimal vertex cover of G_P is a maximal independent set in G_P. Thus, it suffices to compute all maximal independent sets in G_P. Fortunately, an algorithm of computing all maximal independent sets in graphs, with polynomial delay only, has been developed by Johnson et al. [9]. Exploiting this algorithm we use a procedure MinVC(G) to generate all minimal vertex covers of a graph G with polynomial delay. Similarly, we will use a procedure HornSat(H) that returns a minimal model τ of H if and only if H is a satisfiable Horn formula, else returns **nil**, for an appropriate Horn-SAT algorithm see e.g. [11,14]. Now we are ready to state algorithm MHFSAT determining a model τ of $M \in$ MHF, if M is satisfiable, otherwise unsatisfiability (**nil**) of M is reported. For convenience, we identify a vertex cover X of G_P and the corresponding partial model in $M = H \wedge P \in$ MHF. X becomes **nil** if all minimal vertex covers of G_P have been enumerated:

Algorithm (MHFSAT(M, τ)).

`Input:` $\emptyset \neq M \in$ MHF
`Output: model` τ `for` M, `if` $M \in$ SAT, **nil** `otherwise`
begin
`compute` $P := P(M)$
if $P = \emptyset$ **then return** $\tau \leftarrow$ HornSat(M)
`compute graph` G_P
$\tau \leftarrow$ **nil**; $X \leftarrow$ **nil**
repeat
  `compute by` MinVC(G_P)`the next minimal vertex cover` X `of` G_P
  if $X \neq$ **nil then** $\tau \leftarrow$ HornSat($M[X]$)
until $\tau \neq$ **nil or** $X =$ **nil**
return $X \cup \tau$
end

Theorem 2. *Algorithm* MHFSAT *correctly solves the* SAT *search problem in time* $O(2^{0.5284|V(M)|})$, *for* $M \in$ MHF.

Proof. The correctness of the algorithm follows from the argumentation above. Moreover, it is not hard to see that $X =$ **nil** if and only if $\tau =$ **nil** in the last line of the algorithm. Hence, it is ensured that the returned value either is a model for the input instance M or is **nil**.

Addressing the running time, we can compute $P := P(M)$, and the formula graph $G_P = (V(P), P)$ of P in linear time $O(\|M\|)$. If $P = \emptyset$ we are done in linear time by Horn-SAT. If $P \neq \emptyset$ the repeat-until loop is executed. During each iteration we never consume more than the polynomial time delay for computing the next minimal vertex cover followed by a linear time Horn-SAT computation, thus needing only polynomial time. The number of iterations is bounded by the cardinality of minimal vertex covers of G_P. Given a graph G, it is a long standing result by Moon and Moser [15] that the number of its maximal independent sets is bounded by $3^{1/3|V(G)|} \lessapprox 2^{0.5284|V(G)|}$. In fact, this is a tight bound in the sense that there exist graphs achieving this number. Such graphs consist of $n/3$ copies of the K_3, because every triangle independently contributes three different maximal independent sets. Hence, we conclude that SAT for an arbitrary instance $M \in$ MHF is solvable in time $O(p(n)3^{n/3})$ where p denotes an appropriate polynomial, thus providing the claimed time bound of $O(2^{0.5284n})$. □

Let $\mathscr{C} \subseteq$ CNF be a formula class that is closed under formula-reducing, i.e. for $C \in \mathscr{C}$, also $C[\tau] \in \mathscr{C}$, for every truth assignment $\tau : V(C) \to \{0, 1\}$. This requirement is usually satisfied for a class $\mathscr{C}$ unless it is characterized by a minimum clause length greater than 1. Now assume that SAT, for such a class $\mathscr{C}$ is solvable in polynomial time by a procedure $\mathscr{C}$-SAT. Then the proof of Lemma 2 directly applies to formulas $C = D \wedge P \in \mathscr{C} \wedge \mathrm{CNF}_+(=2)$ also. Moreover, replacing subprocedure HornSAT in Algorithm MHFSAT by the subprocedure $\mathscr{C}$-SAT, yields an algorithm solving SAT for the class $\mathscr{C} \wedge \mathrm{CNF}_+(=2)$. Hence, the proof of Theorem 2 also yields:

Theorem 3. *Given $\mathscr{C}$ as above, the* SAT *search problem is solvable in time* $O(2^{0.5284|V(C)|})$, *for* $C \in \mathscr{C} \wedge \mathrm{CNF}_+(=2)$.

Due to Lemma 1, $C \in$ MqHF can be turned in linear time into $\hat{C} \in \mathscr{2H} \wedge$ 2-CNF. The complete 2-CNF part $T(\hat{C})$ of $\hat{C}$ obviously is Horn and thus is in partition form, except for its positive monotone part of 2-clauses, namely $P(\hat{C})$. Thus, we obtain a linear time transformation from C to $\hat{C} \in \mathscr{2H} \wedge \mathrm{CNF}_+(=2)$. Because the SAT search problem can be solved in linear time for a formula in partition-form [2], and because the class $\mathscr{2H}$ obviously is closed under formula-reducing, we obtain, by Theorem 3, the following result (recall MHHF $\subseteq$ MqHF).

Corollary 4. *For $C \in$ MHHF (resp. $C \in$ MqHF), the* SAT *search problem can be solved in time* $O(2^{0.5284|V(C)|})$.

We shall derive another consequence from the preceding discussion. Notice that the variables of $P(M)$, only, are crucial for the running time of Algorithm MHFSAT, because they form the vertex set of the graph $G_{P(M)}$ that has to be investigated.

Corollary 5. *For $M = H \wedge P \in$ MHF, the* SAT *search problem is solvable in time* $O(\|M\| 2^{0.5284|V(P)|})$.

For fixed $k \in \mathbb{N}$, let $\mathrm{MHF}_k := \{M \in \mathrm{MHF} : |V(P(M))| \leqslant k\}$ denote the subclass of MHF where the positive monotone subformulas of 2-clauses have at most k different variables. Similarly, denote by MHHF_k (MqHF_k) the subset of mixed hidden Horn (mixed q-Horn) formulas M whose maximal subformula $T(M) \in$ 2-CNF has at most k different variables. W.r.t. the classes MHF_k, $k \geqslant 0$, we have a fixed-parameter tractability classification of the MHF-SAT problem. By Lemma 1 and Corollary 4, we also obtain w.r.t. the classes MHHF_k (resp. MqHF_k), $k \geqslant 0$, a fixed-parameter tractability classification of the MqHF-SAT problem (recall MHHF $\subset$ MqHF):

Corollary 6. *For $M \in \mathrm{MHF}_k$ (resp. $M \in \mathrm{MHHF}_k$, $M \in \mathrm{MqHF}_k$), $k \geqslant 0$,* SAT *can be decided in polynomial time* $O(\|M\| 2^{0.5284k})$.

For some subclasses of MHF we have slightly better bounds than stated in Corollary 5:

Proposition 7. *Let $M = H \wedge P \in$ MHF with $k = |V(P)|$ and formula graph $G := G_P$ associated to P.*

(1) *There is a polynomial p such that* SAT *can be solved for M in time* $O(p(k)2^{k/2})$ *in either of the following cases*:
  (i) *G is triangle-free.*
  (ii) *G is connected and contains at most one cycle.*
(2) *If G contains at most $r \geqslant 1$ cycles and has at least $3 \cdot r$ vertices, then* SAT *is solvable for M in time* $O(p(k)3^r 2^{(k-3r)/2})$, *for an appropriate polynomial p.*

Proof. It suffices to verify that Algorithm MHFSAT has the claimed running times for the special instances fulfilling the stated properties. Case (1)(i), for G triangle-free, has been solved by Hujter et al. [8], who have shown that a triangle-free graph of at least four vertices contains at most 2^s maximal independent sets if $|V(G)| = 2s$ and at most $5 \cdot 2^{s-2}$ maximal independent sets if $|V(G)| = 2s + 1$. The extremal graphs achieving these bounds consist of s copies of the K_2, respectively, $s - 2$ copies of K_2 and one copy of C_5. Case (ii) was solved by Jou et al. [10]. They have shown that a connected graph with at most one cycle admits at most $3 \cdot 2^{s-2}$ maximal independent sets if $|V(G)| = 2s$ and at most $2^s + 1$ maximal independent sets if $|V(G)| = 2s + 1$. Assertion (2) follows by the above argumentation from the results obtained by Sagan et al. [19]. They have shown that the number of maximal independent sets in graphs containing at most $r \geqslant 1$ (not necessarily non-intersecting) cycles and at least $3r$ vertices is bounded from above by $3^r 2^{(k-3r)/2}$. They also have shown that this bound is tight and is achieved by graphs that consist of copies of an appropriate number of K_3 and K_2. Notice that assertion (2) implies (1)(ii), for $r = 1$ (cf. also [7]). $\square$

5. Hardness of improving Theorem 2

Next we address the question, which improvements of the time bound for solving MHF-SAT presented in Theorem 2 can be expected. To that end we observe a close relationship between MHF-SAT and CNF-SAT:

Theorem 8. *Every formula $C \in$ CNF, in linear time, can be transformed into a SAT-equivalent formula $M_C \in$ MHF such that M_C can be tested for SAT in time $O(p(n)2^{n/2})$, where $n := |V(P(M_C))| \leqslant 2|V(C)|$ and p is an appropriate polynomial.*

Proof. We apply Transformation CNF-To-MHF to an arbitrary formula $C \in$ CNF with the slight modification that also all positive monotone 2-clauses in C (if some exist) are treated in the same way. It is easy to verify that this transformation changes C into a SAT-equivalent formula $M_C \in \mathrm{MHF}_-$ of $n \leqslant 2|V(C)|$ variables such that $G_{P(M_C)}$ consists of isolated edges only. Hence, we obtain the assertion by Proposition 7, (1)(i), and Corollary 5. $\square$

It seems to be very hard to improve on the bound stated in the last theorem significantly, since otherwise SAT for an arbitrary $C \in$ CNF ($n := |V(C)|$) could be solved significantly faster than in 2^n steps. For suppose there is an algorithm solving SAT for MHFs $M = H \wedge P$ with $n = |V(P)|$ in $O(2^{\alpha n})$ steps for some $\alpha < 1/2$. Then we can transform an arbitrary CNF formula C into a SAT-equivalent formula $M_C = H_C \wedge P_C$ with at most $2n$ variables contained in P_C. SAT for M_C, in turn, can be solved in $O(2^{2\alpha n})$ steps, where $2\alpha < 1$. Although, there has been made some progress recently in finding non-trivial bounds for SAT for arbitrary CNF formulas [4], it would require a significant breakthrough in our understanding of SAT to obtain upper time bounds of the form $O(2^{(1-\varepsilon)n})$, for some constant $\varepsilon > 0$.

6. An approach for reducing the number of essential variables

The number of new introduced variables, necessary to transform $C \in$ CNF into $M_C = H_C \wedge P_C \in$ MHF, is crucial regarding the running time of Algorithm MHFSAT. This is due to the fact that these variables contribute vertices to the formula graph of P_C. The requirement to keep this set small leads us to the following notion.

Definition 2. For $C \in$ CNF, a minimal set $X \subseteq V(C)$, for which the transformation in the proof of Theorem 8 yields a MHF formulation $M_C := H_C \wedge P_C \in$ MHF of C via the corresponding set X' of new variables ($|X| = |X'|$), is called an *essential set of variables* (*of C*).

Observe that there may exist many essential sets of variables of a formula C not necessarily of the same cardinality. To obtain the smallest essential set of variables one can proceed as follows: for each clause $c \in C$ that is not Horn, let c' denote the positive monotone part of c. For example $c = \{x, \overline{y}, z\}$ delivers $c' = \{x, z\}$. Collecting these parts c' of all clauses c in C, yields a positive monotone formula $C' \in \mathscr{M}_+$. Resting on the formula graph $G_{C'}$ determined from $C \in$ CNF in that manner, it is not hard to derive the following result.

Lemma 3. *For $C \in$ CNF and C' as defined above, every essential set of variables $X \subset V(C)$ is a minimal vertex cover of the formula graph $G_{C'}$ of C' and vice versa.*

It remains to transform C' into a Horn formula with least effort. For this, we obviously have to search for a *smallest* essential set of variables of a formula $C \in$ CNF, which due to Lemma 3 is a minimum vertex cover of the formula graph $G_{C'}$. A minimum vertex cover of a graph with n vertices can be computed in time $O(2^{n/4})$ by the Robson algorithm [18] determining a maximum independent set.

The preceding argumentation will speed-up Algorithm MHFSAT when applied to MHFs for which sufficiently small essential sets can be found. The idea is to treat a formula $M \in \mathrm{MHF}$ as an input to the transformation just described. It is the part $P := P(M)$ that can be transformed into a Horn formula using a minimum vertex cover of G_P as essential set. Thus, instead of Algorithm MHFSAT we shall proceed by the following Algorithm MHFSAT$*$:

Algorithm (MHFSAT $*$ (M, τ)).

Input: $\emptyset \neq M \in \mathrm{MHF}$, let $P := P(M)$, $k := |V(P)|$
Output: model τ for M, if $M \in \mathrm{SAT}$, **nil** otherwise
begin
compute minimum vertex cover X of G_P by the Robson algorithm
if $|X| < 0.5284 \cdot k$ **then**
  transform M into a new MHF M' using essential set X
  $M \leftarrow M'$
end if
perform Algorithm MHFSAT(M, τ)
end

The Robson algorithm for computing a maximum vertex cover X of G_P runs in time $\mathrm{O}(2^{k/4})$. In case of $|X| \geqslant 0.5284 \cdot k$, we proceed by the usual Algorithm MHFSAT, for the original instance M. Otherwise, i.e. $(*)$: $|X| < 0.5284 \cdot k$, we use X as an essential set of variables for a reformulation of M resulting in a new MHF $M' = P' \wedge H'$, whose positive monotone part P' contains $|V(P')| =: k' = 2|X|$ variables. Moreover, the formula graph $G_{P'}$ by construction consists of isolated edges only (cf. the proof of Theorem 8). Now the computation is completed by applying Algorithm MHFSAT to the modified instance M'. Because of the structure of $G_{P'}$ and according to Proposition 7, (1)(i), we obtain in this branch of Algorithm MHFSAT$*$ the better running time $\mathrm{O}(\|M'\|2^{k'/2}) = \mathrm{O}(\|M'\|2^{|X|})$, where the exponential factor has decreased due to $(*)$.

To illustrate the usefulness of essential sets, again consider the graph coloring problem. Let $C(G)$ be the 3-CNF formula corresponding to the 3-colorability problem of a given graph $G = (V, E)$ as mentioned in the introduction. $C(G)$ consists of $|V|$ positive monotone 3-clauses and $3|E|$ negative monotone 2-clauses and is therefore no MHF formula. Clearly, complementing all variables yields a MHF $H \wedge P$. Unfortunately, the crucial subformula P becomes large by this operation. In order to speed up the SAT test of $H \wedge P$, an essential set of variables in $C(G)$ turning it into a MHF of a smallest P-part is required. As an example, take the triangle graph K_3 with vertex set $\{a, b, c\}$ leading to the CNF formula $C(G) = C(V) \cup C(E)$ with corresponding clause sets:

$$C(V) := \{\{a_1, a_2, a_3\}, \{b_1, b_2, b_3\}, \{c_1, c_2, c_3\}\},$$

$$C(E) := \{\{\overline{a}_1, \overline{b}_1\}, \{\overline{a}_2, \overline{b}_2\}, \{\overline{a}_3, \overline{b}_3\}\} \cup \{\{\overline{a}_1, \overline{c}_1\}, \{\overline{a}_2, \overline{c}_2\}, \{\overline{a}_3, \overline{c}_3\}\} \cup \{\{\overline{b}_1, \overline{c}_1\}, \{\overline{b}_2, \overline{c}_2\}, \{\overline{b}_3, \overline{c}_3\}\}.$$

Turning this into a MHF by complementing all variables yields a P-subformula of 9 clauses and 18 variables. Taking instead only an essential set of 6 variables, namely 2 variables of each 3-clause in $C(V)$, e.g. $\{a_i, b_i, c_i : i = 1, 2\}$, yields a MHF $M(G) = H \wedge P$ with

$$P := \{\{a_1, a'_1\}, \{a_2, a'_2\}, \{b_1, b'_1\}, \{b_2, b'_2\}, \{c_1, c'_1\}, \{c_2, c'_2\}\},$$

$$H := C(E) \cup P^{\gamma} \cup \{\{\overline{a'}_1, \overline{a'}_2, a_3\}, \{\overline{b'}_1, \overline{b'}_2, b_3\}, \{\overline{c'}_1, \overline{c'}_2, c_3\}\}.$$

Recall that C^{γ} means to complement all variables in formula C. The new formula P contains only 6 clauses and only 12 variables instead of 18, moreover the formula graph consists of isolated edges only. Although the example is simple, it describes the usefulness of essential sets, which becomes explicit when dealing with larger instances.

To supply these observations we had run several experiments for CNF(=3) formulas with 1000 variables and $c \cdot 1000$ clauses, for $c = 1, \ldots, 6$. Each $C \in \mathrm{CNF}(=3)$ has been generated randomly and was transformed into $M_C = H \wedge P$. The new introduced variables form an essential set of variables of C. Fig. 1 displays the average number of essential

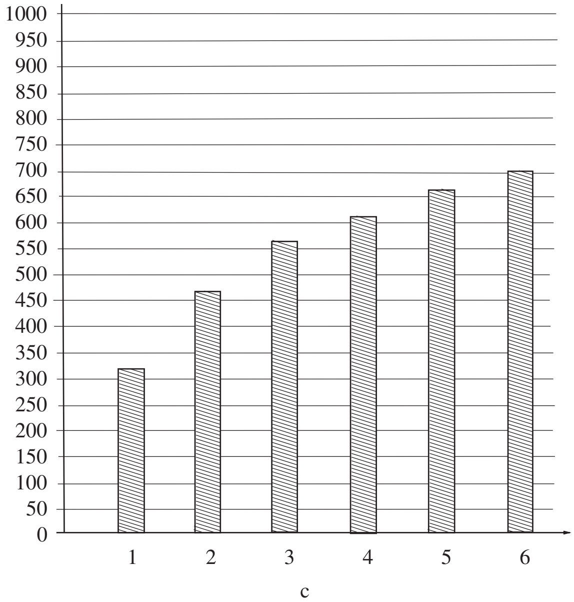

Fig. 1. Number of essential variables in MHFs with 1000 variables and $c \cdot 1000$ clauses ($c = 1, \dots, 6$).

variables, computed by a maximum-degree greedy vertex-cover heuristic, obtained from samples of 100 formulas, for each $c = 1, \dots, 6$.

7. Solving minimum weight MHF-SAT in time $O(2^{0.5284n})$

In the following we are interested in the minimum weight satisfiability problem (MINV-SAT) where weights are assigned to variables. For $C \in$ CNF and weight function $w : V(C) \to \mathbb{R}$, MINV-SAT searches for a *minimum model* for C, if existing, i.e. a model of minimal weight. The weight of a model τ is defined as $w(\tau) := \sum_{x \in \tau_T} w(x) = \sum_{x \in V(C)} \tau(x) w(x)$. Here, $\tau_T := \tau^{-1}(1)$ denotes the set of variables in $V(C)$ assigned to 1 by τ. Note that for a constant weight function $w = c$, MINV-SAT means to find a model τ such that $|\tau_T|$ is minimal. We have analogous definitions for the maximum version of the problem, called MAXV-SAT. Observe that MINV-SAT is an optimization problem which is NP-complete in its decision version even if non-negative weights, only, are assigned to variables and we restrict to the class 2-CNF. This follows by a straightforward reduction from minimum weight vertex cover to MINV-SAT, for the subclass $\mathrm{CNF}_{+}(=2)$. The same reduction, using the class $\mathrm{CNF}_{-}(=2)$, shows that MAXV-SAT is NP-hard, too.

For a truth assignment $\tau : V \to \{0, 1\}$, we denote by τ^{γ} the truth assignment obtained from τ by flipping all assignments. Given an integer weight function w let $-w$ denote the function obtained from w by multiplying each of its values by -1. We state some useful observations:

Lemma 4. (1) *For $C \in$ CNF and $w : V(C) \to \mathbb{R}$ holds that τ is a minimum weight model for (C, w) if and only if τ^{γ} is a minimum weight model for $(C^{\gamma}, -w)$.*

(2) *Let $\mathscr{C} \subseteq$ CNF over variable set V be a formula class for which minimum weight* SAT *can be solved in time* $O(f(|V|))$, *if non-negative weights are assigned to the variables. Then also maximum weight* SAT *for $\mathscr{C}$ is solvable in time* $O(f(|V|))$ *if non-positive weights are assigned to the variables.*

Proof. For proving (1) first observe that τ obviously is a model of C iff τ^{γ} is a model of C^{γ}, and $V(C) = V(C^{\gamma})$. Moreover, one easily verifies that $(*) : w(\tau^{\gamma}) + w(\tau) = w(V(C)) = \sum_{x \in V(C)} w(x)$ holds, for every truth assignment $\tau : V(C) \to \{0, 1\}$. Now let τ be a minimum weight model of (C, w) but suppose τ^{γ} is not a minimum weight model of (C^{γ}, w'), where $w' := -w$. Then there exists a model τ_0 of C^{γ} such that $w'(\tau_0) < w'(\tau^{\gamma})$ equivalent to $w(\tau_0) > w(\tau^{\gamma})$. Because τ_0^{γ} is a model of C we have $w(\tau_0) = w(V(C)) - w(\tau_0^{\gamma})$ due to $(*)$. Comparing the last two relations and again using $(*)$ yields $w(\tau_0^{\gamma}) < w(\tau)$ contradicting that τ is a minimum weight model of (C, w). Proving the converse direction proceeds analogously.

Addressing (2) first observe that $\max_{\tau\in\mathscr{T}(C)} w(\tau) = -\min_{\tau\in\mathscr{T}(C)}(-w(\tau))$, for $C \in \mathscr{C}$, $w : V(C) \to \mathbb{R}_-$, and $\mathscr{T}(C)$ denoting the set of all models of C. Therefore, the weight of a maximum weight model of (C, w) can be computed via the weight of a minimum weight model of $(C, -w)$. To prove the assertion for the optimal models itself we claim that $\mathscr{T}_{\min}(C, w) = \mathscr{T}_{\max}(C, -w)$ denoting the sets of minimum resp. maximum weight models, respectively, from which the assertion follows. To verify the claim let $\tau \in \mathscr{T}_{\min}(C, w)$, and assume $\tau \notin \mathscr{T}_{\max}(C, w')$ where $w' := -w$. Then there exists $\tau_0 \in \mathscr{T}(C)$ with $w'(\tau_0) > w'(\tau)$ which is equivalent to $-w'(\tau_0) < -w'(\tau)$ meaning $w(\tau_0) < w(\tau)$ contradicting $\tau \in \mathscr{T}_{\min}(C, w)$. Therefore, $\mathscr{T}_{\min}(C, w) \subseteq \mathscr{T}_{\max}(C, -w)$. Analogously, we obtain $\mathscr{T}_{\max}(C, -w) \subseteq \mathscr{T}_{\min}(C, w)$. □

Next we show that a modified version of Algorithm MHFSAT also serves for solving MINV-SAT for the class MHF when arbitrary non-negative weights are assigned to the variables. Let us mention a useful observation for MINV-HornSAT, i.e. MINV-SAT restricted to Horn formulas. This observation is based on a procedure HornSAT that solves the usual SAT problem for a Horn formula H in linear time, i.e. it finds a model if and only if H is satisfiable, else returns FALSE, see e.g. [14,11].

Lemma 5. *Procedure HornSAT, in linear time, finds a minimum weight model for a satisfiable Horn formula $H \in \mathscr{H}$ and weight function $w : V(H) \to \mathbb{R}_+$, else returns nil.*

Proof. By definition a variable is set to 1 by procedure HornSAT only if it is necessary, all other variables are set to 0. This means HornSAT finds the *unique* minimal model of a Horn formula if a model exists for it, at all. Since every minimum model must be a minimal model, we are done. □

Now we are ready to present Algorithm MINV-MHFSAT determining a minimum weight model τ for $M \in$ MHF, $w : V(M) \to \mathbb{R}_+$, if M is satisfiable, otherwise unsatisfiability (**nil**) of M is reported. For convenience we, again, identify a vertex cover X of G_P and the corresponding partial model in $M = H \wedge P \in$ MHF. X becomes **nil** if all minimal vertex covers of G_P have been enumerated.

Algorithm (MINV-MHFSAT(M, τ)).

```
Input: ∅ ≠ M ∈ MHF, w : V(M) → ℝ+
Output: minimum model τ for M, if M ∈ SAT, nil otherwise
begin
compute P := P(M)
if P = ∅ then return τ ← HornSat(M)
compute graph G_P
τ ← nil; X ← nil; w(τ) ← ∞
repeat
  compute by MinVC(G_P) the next minimal vertex cover X of G_P
  if X ≠ nil then
    σ ← HornSat(M[X])
    if σ ≠ nil then
      if w(X) + w(σ) < w(τ) then τ ← X ∪ σ
    end if
  end if
until X= nil
if σ= nil then return nil
else return τ
end
```

Theorem 9. *Algorithm* MINV-MHFSAT *solves minimum weight* SAT *in time* $O(2^{0.5284|V(M)|})$, *for arbitrary* $M \in$ MHF, $w : V(M) \to \mathbb{R}_+$.

Proof. To establish correctness and completeness of the algorithm, we first observe that it returns **nil** if and only if the input formula is not satisfiable. Moreover, like in the proof of Theorem 2 it follows that if the algorithm returns a truth assignment, then it is a correct model of M.

To finish the correctness proof, assume that the algorithm returns a model τ for M that is not optimal. Then, as $M \in$ SAT, there must exist a model τ^* of smaller weight: $w(\tau^*) < w(\tau)$. Clearly, the restriction $\tau^*|V(P)$ contains a minimal vertex cover X of G_P (given by corresponding variables that are assigned to 1 by τ^*) otherwise it is no model of M. Because X is contained in the set of variables set to 1 by τ, we get the composition $w(\tau^*) = w(X) + w(\tau^*|V(M[X]))$; recall that $M[X]$ is obtained by reducing M according to setting all variables in X to 1. On the other hand, Algorithm MINV-MHFSAT must have considered also the minimal vertex cover X in its repeat-until-loop. As the remaining formula $M[X]$ is a Horn formula, the algorithm by HornSAT finds the unique minimum model σ^* for $M[X]$ due to Lemma 5. Hence, $w(\tau) = w(X) + w(\sigma^*)$, and therefore we obtain $w(\sigma^*) > w(\tau^*|V(M[X]))$ contradicting the optimality of σ^*. We conclude that Algorithm MINV-MHFSAT always correctly outputs an optimal model if one exists, else correctly reports that there is none.

The running time obviously is the same as that of Algorithm MHFSAT, hence we are done by Theorem 2. $\square$

In Section 5 strong arguments are provided that it seems to be very hard to improve on the time bound of $O(2^{0.5284n})$ for MHF-SAT. By the same argumentation it follows that improving the presented bound for MINV-MHFSAT is even harder. Notice, further, that Algorithm MINV-MHFSAT, in general, does not work for arbitrary real weights assigned to the variables. For this, we had to generate all vertex covers of G_P not only the minimal ones and for the remaining Horn formula we could not simply proceed by HornSAT: Assume all weights are -1, then we had to find a maximum Horn model, which is an NP-hard optimization problem. This follows easily by reduction from the Minimum Hypergraph Transversal (or Minimum Hitting Set) problem, based on the class of negative monotone Horn formulas.

Let $\mathrm{MHF}^\gamma = \{M^\gamma : M \in \mathrm{MHF}\}$ denote the class of *dual* mixed Horn formulas. Because of Lemma 4(1) it follows from Theorem 9 that we can also solve MINV-SAT, for MHF^γ. Additionally referring to Lemma 4(2), we have:

Corollary 10. MAXV-SAT *for* MHF (*resp.* MINV-SAT *for* MHF^γ) *is solvable in time* $O(2^{0.5284n})$, *for formulas over variable set* V, $|V| = n$ *and* $w : V \to \mathbb{R}_-$. *Moreover*, MAXV-SAT *for* MHF^γ *is solvable in time* $O(2^{0.5284n})$ *for instances over variable set* V, $|V| = n$ *and* $w : V \to \mathbb{R}_+$.

Further, let us state some fixed-parameter tractability classifications following directly from the results stated above.

Corollary 11. MINV-SAT (*resp.* MAXV-SAT) *is fixed-parameter tractable in time* $O(\|M\| 2^{0.5284k})$, *for the following classes*:

(1) $M \in \mathrm{MHF}_k$ *and* $w : V(M) \to \mathbb{R}_+$ (*resp.* $w : V(M) \to \mathbb{R}_-$).
(2) $M \in \mathrm{MHF}_k^\gamma$ *and* $w : V(M) \to \mathbb{R}_-$ (*resp.* $w : V(M) \to \mathbb{R}_+$).

As 2-CNF $\subset \mathrm{MHF} \cap \mathrm{MHF}^\gamma$ we finally obtain:

Corollary 12. MINV-SAT (*respectively* MAXV-SAT) *can be solved in time* $O(2^{0.5284|V(C)|})$, *for formulas* $C \in$ 2-CNF *and* $w : V(C) \to \mathbb{R}_+$ (*resp.* $w : V(C) \to \mathbb{R}_-$).

It is an open problem to find an algorithm solving minimum weight SAT for 2-CNF running faster than Algorithm MINV-MHFSAT.

References

[1] B. Aspvall, M.R. Plass, R.E. Tarjan, A linear-time algorithm for testing the truth of certain quantified Boolean formulas, Inform. Process. Lett. 8 (1979) 121–123.
[2] E. Boros, Y. Crama, P.L. Hammer, Polynomial time inference of all valid implications for Horn and related formulae, Ann. Math. Artif. Intell. 1 (1990) 21–32.
[3] E. Boros, P.L. Hammer, X. Sun, Recognition of q-Horn formulae in linear time, Discrete Appl. Math. 55 (1994) 1–13.
[4] E. Dantsin, A. Wolpert, Algorithms for SAT based on search in Hamming balls, ECCC Report No. 17, 2004.

[5] R.G. Downey, M.R. Fellows, Parameterized Complexity, Springer, New York, 1999.
[6] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, San Francisco, 1979.
[7] J.R. Griggs, C.M. Grinstead, D.R. Guichard, The number of maximal independent sets in a connected graph, Discrete Math. 68 (1988) 211–220.
[8] M. Hujter, Z. Tuza, The number of maximal independent sets in triangle-free graphs, SIAM J. Discrete Math. 6 (1993) 284–288.
[9] D.S. Johnson, M. Yannakakis, C.H. Papadimitriou, On generating all maximal independent sets, Inform. Process. Lett. 27 (1988) 119–123.
[10] M. Jou, G.J. Chang, Maximal independent sets in graphs with at most one cycle, Discrete Appl. Math. 79 (1997) 67–73.
[11] H. Kleine Büning, T. Lettman, Propositional Logic, Deduction and Algorithms, Cambridge University Press, Cambridge, 1999.
[12] D.E. Knuth, Nested satisfiability, Acta Inform. 28 (1990) 1–6.
[13] H.R. Lewis, Renaming a set of clauses as a Horn set, J. ACM 25 (1978) 134–135.
[14] M. Minoux, LTUR: a simplified linear-time unit resolution algorithm for Horn formulae and computer implementation, Inform. Process. Lett. 29 (1988) 1–12.
[15] J.W. Moon, L. Moser, On cliques in graphs, Israel J. Math. 3 (1965) 23–28.
[16] S. Porschen, E. Speckenmeyer, Worst case bounds for some NP-complete modified Horn-SAT problems, in: Proceedings of the 7th International Conference on Theory and Applications of Satisfiability Testing (SAT'04), Vancouver, British Columbia, Canada, Lecture Notes in Computer Science, vol. 3542, Springer, Berlin, 2005, pp. 251–262.
[17] B. Randerath, E. Speckenmeyer, E. Boros, P. Hammer, A. Kogan, K. Makino, B. Simeone, O. Cepek, A Satisfiability formulation of problems on level graphs, ENDM 9 (2001).
[18] J.M. Robson, Finding a maximum independent set in time $O(2^{n/4})$, ⟨http://dept-info.labri.u-bordeaux.fr/∼robson/mis/techrep.html⟩, 2001.
[19] B.E. Sagan, V.R. Vatter, Maximal and maximum independent sets in graphs with at most *r* cycles, preprint 2003, arXiv:math CO/0207100 v2.
[20] H. van Maaren, L. van Norden, Hidden threshold phenomena for fixed-density SAT-formulae, in: E. Giunchiglia, A. Tacchella (Eds.), Proceedings of the 6th International Conference on Theory and Applications of Satisfiability Testing (SAT'03), Santa Margherita Ligure, Italy, Lecture Notes in Computer Science, vol. 2919, Springer, Berlin, 2004, pp. 135–149.